NEW DIRECT MULTICHANNEL ACTIVE NOISE CONTROL BY FREQUENCY-DOMAIN APPROACH

Masaki Kamata, and Akira Sano

Department of System Design Engineering, Keio University
3-14-1 Hiyoshi, Kohoku-ku, Yokohama, 223-8522, Japan
phone: + (81) 45 566 1730, fax: + (81) 566 1720, email: sano@sd.keio.ac.jp

ABSTRACT

A new frequency-domain direct adaptive approach is proposed for general multichannel active noise control (ANC) when both of the primary and secondary path channels are uncertain and changeable. To reduce the cancelling errors, two kinds of virtual error vectors are introduced and are forced to zero by adjusting three adaptive FIR filter matrices in an online manner, by which the convergence of the actual cancelling error to zero can be attained at the objective points. Unlike other conventional approaches, the proposed algorithm can give an inverse controller directly without need of explicit identification of the secondary path channels. The proposed algorithm can be implemented in the frequency-domain to reduce its computational complexity.

1. INTRODUCTION

Active noise control (ANC) is a way of suppressing unwanted low frequency noises generated by primary sound sources by emitting artificial secondary sounds to the objective points. Since the path dynamics cannot be precisely modeled and may be uncertain and changeable, adaptive tuning of the inverse controller is needed for ANC. A variety of filtered-x LMS algorithms have been proposed to attain the cancellation via feedforward adaptation. Modifications of the filtered-x algorithm have also been investigated from the stability point of view and stability assured filtered-x algorithms have been given by one of the authors [1, 2, 3]. Stability assured filtered-x algorithms have also been investigated in [4, 5].

To deal with a general case when the secondary path channels are unknown or changeable, there have been two adaptive approaches: One is an indirect adaptive approach which is based on on-line identification of the secondary path dynamics, and the filtered-x algorithm using the identified model, and the other is a direct adaptive approach which can directly tune an adaptive feedforward controller without explicit identification of the path channels. In the indirect adaptive approaches, the identified model updates the secondary path model appearing in the filtered-x algorithms. The stability of the filtered-x algorithm linked with the secondary path identification is not assured. According to the identified secondary path models, the feedforward inverse controller can also be redesigned in a real-time manner. On the other hand, no efficient direct adaptive approaches have been proposed to deal with a general case in which all the path channel matrices are unknown [6]. Thus, the proposed algorithm is based on the direct algorithm.

Block frequency-domain implementations of adaptive filters can considerably improve their computational efficiency. In particular, the frequency-domain LMS algorithm represents an exact and efficient implementation of the block time-domain LMS algorithm. The greatest computational savings can be obtained for long adaptive filter lengths, by performing both the convolution involved in the filtering operation, and the correlation involved in the adaptation process, in the frequency-domain using the fast Fourier transform (FFT). Suitable precautions then have to be taken to avoid circular convolution and correlation effects, such as the use of the overlap-save method, and the use of causal gradient constraints, as reviewed, for example, by [7].

Therefore, the purpose of this paper is to propose a new direct adaptive approach to a general multichannel case in frequency-domain, which does not require explicit identification of the secondary path dynamics. To reduce the canceling errors, two virtual error vectors are introduced and forced to zero by adjusting parameters in three adaptive filter matrices in an online manner.

2. DIRECT ADAPTIVE APPROACH

2.1 Multichannel ANC Problem

Fig.1 shows an equivalent structure of multichannel feedforward ANC system. The signal $\boldsymbol{r}(n) \in \mathscr{R}^{N_r}$ detected by N_r reference microphones are the inputs to $N_c \times N_r$ adaptive feedforward controller matrix $\hat{\boldsymbol{C}}(z,k)$, where N_c is the number of the secondary loudspeakers which produce artificial sounds $\boldsymbol{u}(n) \in \mathscr{R}^{N_c}$ to cancel the primary source noise at the N_e objective points. The canceling errors are detected as $\boldsymbol{e}(n) \in \mathscr{R}^{N_e}$ by the N_e error microphones. $\boldsymbol{H}(z) \in \mathscr{L}^{N_e \times N_r}$ and $\boldsymbol{G}(z) \in \mathscr{L}^{N_e \times N_c}$ are the equivalent primary and secondary path matrices respectively. Thus the canceling error $\boldsymbol{e}(n)$ is expressed in terms of the accessible signals $\boldsymbol{r}(n)$ and $\boldsymbol{u}(n)$, as

$$\boldsymbol{e}(n) = \boldsymbol{H}(z)\boldsymbol{r}(n) - \boldsymbol{G}(z)\boldsymbol{u}(n) \tag{1}$$

where $\boldsymbol{H}(z)$ and $\boldsymbol{G}(z)$ are referred to as the equivalent primary and secondary channel matrices respectively. In the active noise control, we cannot measure the signals $\boldsymbol{d}(n)$ and $\boldsymbol{y}(n)$ separately, but can only measure the canceling error $\boldsymbol{e}(n)$, since the model of $\boldsymbol{G}(z)$ involves uncertainty. Thus, the multichannel active noise control problem is how to tune the controller $\boldsymbol{C}(z)$ using only accessible signals $\boldsymbol{r}(n)$, $\boldsymbol{u}(n)$ and $\boldsymbol{e}(n)$, even if the sound transmission matrices $\boldsymbol{H}(z)$ and $\boldsymbol{G}(z)$ are uncertain.

2.2 Time-domain adaptive algorithm

Fig.1 shows a new direct adaptive tuning algorithm for a multichannel ANC in time-domain. We introduce two kinds of virtual error vectors $\boldsymbol{e}_A(n)$ and $\boldsymbol{e}_B(n)$, which are forced

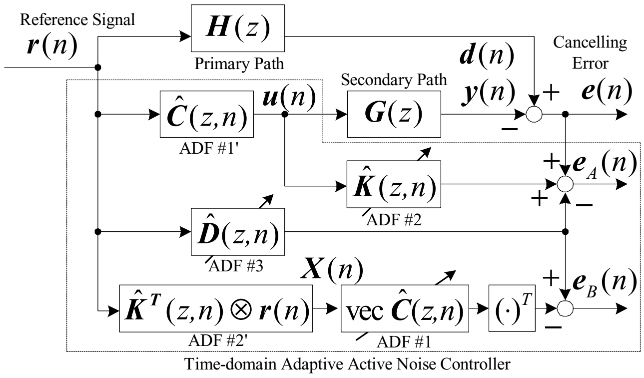


Figure 1: Time-domain algorithm for multichannel ANC

to zero by using three adaptive FIR matrix filters $\hat{\boldsymbol{C}}(z,n)$, $\hat{\boldsymbol{K}}(z,n)$ and $\hat{\boldsymbol{D}}(z,n)$. Thus we can give the expression of the virtual errors as:

$$\boldsymbol{e}_A(n) = \boldsymbol{e}(n) + \hat{\boldsymbol{K}}(z,n)\boldsymbol{u}(n) - \hat{\boldsymbol{D}}(z,n)\boldsymbol{r}(n) \quad (2)$$

$$\boldsymbol{e}_B(n) = \hat{\boldsymbol{D}}(z,n)\boldsymbol{r}(n) - [vec[\hat{\boldsymbol{C}}(z,n)]\boldsymbol{X}(n)]^T \quad (3)$$

where $\boldsymbol{u}(n) = \hat{\boldsymbol{C}}(z,n)\boldsymbol{r}(n)$, $\boldsymbol{X}(n) = \hat{\boldsymbol{K}}^T(z,n) \otimes \boldsymbol{r}(n)$ and $vec[\boldsymbol{A}]$ denotes a row vector expansion of a matrix $\boldsymbol{A}$, and $\otimes$ denotes the Kronecker product.

Then we consider the sum of two virtual errors in Fig.1 from (2) and (3) as

$$\boldsymbol{e}_A(n) + \boldsymbol{e}_B(n) = \boldsymbol{e}(n) + \hat{\boldsymbol{K}}(z,n)\boldsymbol{u}(n) - [vec[\hat{\boldsymbol{C}}(z,n)]\boldsymbol{X}(n)]^T$$

If the coefficient parameters in the three adaptive FIR filters $\hat{\boldsymbol{C}}(z,n)$, $\hat{\boldsymbol{K}}(z,n)$ and $\hat{\boldsymbol{D}}(z,n)$ can be updated so that the error vectors $\boldsymbol{e}_A(n)$ and $\boldsymbol{e}_B(n)$ may become zero, and the filter parameters converge to any constant values, we can show that the canceling error $\boldsymbol{e}(n)$ can also converge to zero. It can be verified by proving that

$$\hat{\boldsymbol{K}}(z,n)\hat{\boldsymbol{C}}(z,n)\boldsymbol{r}(n) = [vec[\hat{\boldsymbol{C}}(z,n)]\hat{\boldsymbol{K}}^T(z,n) \otimes \boldsymbol{r}(n)]^T \quad (4)$$

in sufficiently large n. If the parameters in the all adaptive filters converge to constants, we can exchange the product of two polynomials $\hat{C}_{ij}(z)$ and $\hat{K}_{mi}(z)$ in (4), and then we can establish that $\boldsymbol{e}_A(n) + \boldsymbol{e}_B(n) = \boldsymbol{e}(n)$ in sufficiently large time. Thus, we can assure the convergence of $\boldsymbol{e}(n)$ to zero through the convergence of $\boldsymbol{e}_A(n)$ and $\boldsymbol{e}_B(n)$ to zero.

2.3 Adaptation for Time-domain technique

We express the three adaptive filters $\hat{\boldsymbol{C}}(z,n)$, $\hat{\boldsymbol{K}}(z,n)$ and $\hat{\boldsymbol{D}}(z,n)$ as:

$$\hat{C}_{ij}(z,n) = \hat{c}_{ij}^{(1)}(n)z^{-1} + \hat{c}_{ij}^{(2)}(n)z^{-2} + \cdots + \hat{c}_{ij}^{(L_{ij}^C)}(n)z^{-L_{ij}^C}$$

$$\hat{K}_{mi}(z,n) = \hat{k}_{mi}^{(1)}(n)z^{-1} + \hat{k}_{mi}^{(2)}(n)z^{-2} + \cdots + \hat{k}_{mi}^{(L_{mi}^K)}(n)z^{-L_{mi}^K}$$

$$\hat{D}_{mj}(z,n) = \hat{d}_{mj}^{(1)}(n)z^{-1} + \hat{d}_{mj}^{(2)}(n)z^{-2} + \cdots + \hat{d}_{mj}^{(L_{mj}^D)}(n)z^{-L_{mj}^D}$$

where $i = 1,\cdots,N_c$, $j = 1,\cdots,N_r$, and $m = 1,\cdots,N_e$.

It follows from Fig.1 that the first virtual error vector $\boldsymbol{e}_A$ is expressed by:

$$e_{A,m}(n) = e_m(n) + \Sigma_{i=1}^{N_c}\hat{K}_{mi}(z,n)u_i(n) - \Sigma_{j=1}^{N_r}\hat{D}_{mj}(z,n)r_j(n)$$

$$= e_m(n) + \Sigma_{i=1}^{N_c}\hat{\boldsymbol{\omega}}_{mi}^T(n)\hat{\boldsymbol{\theta}}_{K,mi}(n) - \Sigma_{i=1}^{N_r}\hat{\boldsymbol{\xi}}_{mj}^T(n)\hat{\boldsymbol{\theta}}_{D,mj}(n)$$

where $\omega_{mi}(n) = [u_i(n-1),\cdots,u_i(n-L_{mi}^K)]^T$, $\boldsymbol{\xi}_{mj}(n) = [r_j(n-1),\cdots,r_j(n-L_{mi}^D)]^T$, $\hat{\boldsymbol{\theta}}_{K,mi}(n) = [\hat{k}_{mi}^{(1)}(n),\cdots,\hat{k}_{mi}^{(L_{mi}^{(K)})}(n)]^T$, and $\hat{\boldsymbol{\theta}}_{D,mj}(n) = [\hat{d}_{mj}^{(1)}(n),\cdots,\hat{d}_{mj}^{(L_{mj}^{(D)})}(n)]^T$.

Then from the minimization of the instantaneous squared error norm $||\boldsymbol{e}_A(n)||^2$ with respect to $\hat{\boldsymbol{\theta}}_{K,mi}(n)$ and $\hat{\boldsymbol{\theta}}_{D,mj}(n)$, we can derive the adaptive algorithm for updating these parameters as follows:

$$\hat{\boldsymbol{\theta}}_{K,mi}(n+1) = \hat{\boldsymbol{\theta}}_{K,mi}(n) - \gamma(n)\boldsymbol{\omega}_{mi}(n)e_{A,m}(n) \quad (5)$$

$$\hat{\boldsymbol{\theta}}_{D,mj}(n+1) = \hat{\boldsymbol{\theta}}_{D,mj}(n) + \gamma(n)\boldsymbol{\xi}_{mj}(n)e_{A,m}(n) \quad (6)$$

$$\gamma(n) = \frac{2\alpha||\boldsymbol{e}_A(n)||^2}{\rho + \Sigma_{m=1}^{N_e}e_{A,m}^2(n)(||\boldsymbol{\omega}_m(n)||^2 + ||\boldsymbol{\xi}_m(n)||^2)}$$

where $\boldsymbol{\omega}_m(n) = [\boldsymbol{\omega}_{m1}^T(n),\cdots,\boldsymbol{\omega}_{mN_c}^T(n)]^T$, $\boldsymbol{\xi}_m(n) = [\boldsymbol{\xi}_{m1}^T(n),\cdots,\boldsymbol{\xi}_{mN_r}^T(n)]^T$, and $0 < \alpha < 1, \rho > 0$ is a small constant. The algorithm (5) and (6) have a feature that the step size is not constant but is adjusted by the error vector $\boldsymbol{e}_A(n)$.

On the other hand, the second virtual error is given by:

$$\begin{aligned} e_{B,m}(n) &= \Sigma_{j=1}^{N_r}\hat{\boldsymbol{D}}_{mj}(z,n)r_j(n) \\ &\quad -[\hat{\boldsymbol{C}}_{11}(z,n),\cdots,\hat{\boldsymbol{C}}_{N_cN_r}(z,n)]\cdot[x_{m11}(n),\cdots,x_{mN_cN_r}(n)]^T \\ &= \Sigma_{j=1}^{N_r}\hat{\boldsymbol{D}}_{mj}(z,n)r_j(n) \\ &\quad -(\boldsymbol{x}_{m11}^T(n),\cdots,\boldsymbol{x}_{mN_cN_r}^T(n))\cdot[\hat{\boldsymbol{c}}_{11}(n),\cdots,\hat{\boldsymbol{c}}_{N_cN_r}(n)]^T \\ &= \Sigma_{j=1}^{N_r}\hat{\boldsymbol{D}}_{mj}(z,n)r_j(n) - \boldsymbol{\phi}_{X,m}^T(n)\hat{\boldsymbol{\theta}}_C(n) \end{aligned}$$

where $\boldsymbol{x}_{mij}(n) = [x_{mij}(n-1),\cdots,x_{mij}(n-L_{ij}^C)]^T$, $\hat{\boldsymbol{c}}_{ij}(n) = [\hat{c}_{ij}^{(1)}(n),\cdots,\hat{c}_{ij}^{(L_{ij}^C)}(n)]^T$, $\hat{\boldsymbol{\theta}}_C = [\hat{\boldsymbol{c}}_{11}^T(n),\cdots,\hat{\boldsymbol{c}}_{1N_r}^T,\cdots,\hat{\boldsymbol{c}}_{N_cN_r}^T(n)]^T$, $\boldsymbol{\phi}_{X,m}^T(n) = [\boldsymbol{x}_{m11}^T(n),\cdots,\boldsymbol{x}_{m1N_r}^T,\cdots,\boldsymbol{x}_{mN_cN_r}^T(n)]$.

Thus, the second virtual error vectors are expressed by:

$$\boldsymbol{e}_B(n) = \hat{\boldsymbol{D}}(z,n)\boldsymbol{r}(n) - \boldsymbol{\Phi}_X^T(n)\hat{\boldsymbol{\theta}}_C(n)$$

where $\boldsymbol{\Phi}_X(n) \equiv [\boldsymbol{\phi}_{X,1}(n),\boldsymbol{\phi}_{X,2}(n),\cdots,\boldsymbol{\phi}_{X,N_e}(n)]$. Then, we can give the adaptive algorithm for updating the parameters in $\hat{\boldsymbol{C}}(z,n)$ as follows:

$$\hat{\boldsymbol{\theta}}_C(n+1) = \hat{\boldsymbol{\theta}}_C(n) + \gamma_c(n)\boldsymbol{\Phi}_X(n)\boldsymbol{e}_B(n) \quad (7)$$

$$\gamma_c(n) = \frac{2\alpha||\boldsymbol{e}_B(n)||^2}{\rho_c + ||\boldsymbol{\Phi}_X(n)\boldsymbol{e}(n)||^2}$$

where $0 < \alpha_c < 1$, and $\rho > 0$ is a small constant.

Then by updating the old parameters of $\hat{\boldsymbol{\theta}}_C(n)$ and $\hat{\boldsymbol{\theta}}_K(n)$ in ADF#$1'$ and ADF#$2'$ in Fig.1 by the new adjusted parameters in (5), (6) and (7), we can generate the control inputs $\boldsymbol{u}(n)$ and the auxiliary signals $\boldsymbol{X}(n)$.

3. FREQUENCY-DOMAIN IMPLEMENTATION

3.1 Expression of two virtual error vectors

Fig.2 shows a new fully adaptive tuning algorithm in the frequency-domain for a multichannel ANC. We introduce two kinds of virtual error vectors $\boldsymbol{e}_A(k)$ and $\boldsymbol{e}_B(k)$, which are forced to zero by using three frequency-domain adaptive FIR filter matrices $\tilde{\boldsymbol{C}}(k)$, $\tilde{\boldsymbol{K}}(k)$, and $\tilde{\boldsymbol{D}}(k)$.

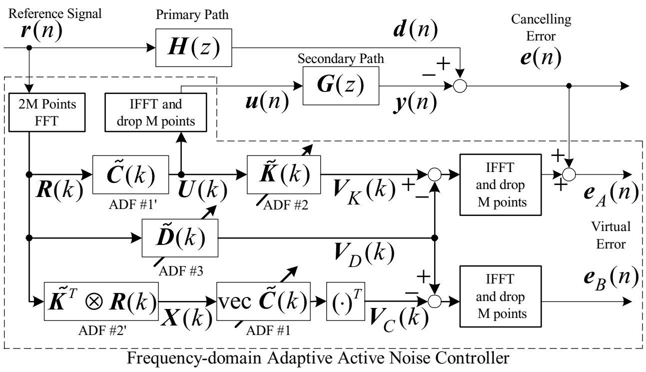


Figure 2: Freq.-domain algorithm for multichannel ANC

Let k refer to the block index, which is related to the original sample time n as: $n = kM + s,\ \ s = 0, 1, \cdots, M-1,\ \ k = 1, 2, \cdots$, where M is the block length.

The reference data which is detected in j-th reference microphone for block k is thus defined by the set $\{\boldsymbol{r}_j(kM + s)\}_{s=0}^{M-1}$, which is written in a vector form as follows:

$$\boldsymbol{r}_j(k) = [r_j(kM), r_j(kM+1) \cdots, r_j(kM+M-1)]^T$$

The j-th frequency-domain reference vector $\boldsymbol{R}_j(k)$ of size $N = 2M$ are calculated applying FFT's on the corresponding time-domain vectors as

$$\boldsymbol{R}_j(k) = diag\{FFT[\boldsymbol{r}_j(k-1)^T, \boldsymbol{r}_j(k)^T]\}$$

Correspondingly, we have introduced three adaptive filter $\tilde{\boldsymbol{c}}_{ij}(k)$,$\tilde{\boldsymbol{k}}_{mi}(k)$, and $\tilde{\boldsymbol{d}}_{mj}(k)$ as follows:

$$\begin{aligned}
\tilde{\boldsymbol{c}}_{ij}(k) &= [\tilde{c}_{ij}^{(0)}(k), \tilde{c}_{ij}^{(1)}(k), \cdots, \tilde{c}_{ij}^{(N-1)}(k)] \\
\tilde{\boldsymbol{k}}_{mi}(k) &= [\tilde{k}_{mi}^{(0)}(k), \tilde{k}_{mi}^{(1)}(k), \cdots, \tilde{k}_{mi}^{(N-1)}(k)] \\
\tilde{\boldsymbol{d}}_{mj}(k) &= [\tilde{d}_{mj}^{(0)}(k), \tilde{d}_{mj}^{(1)}(k), \cdots, \tilde{d}_{mj}^{(N-1)}(k)]
\end{aligned}$$

where $i = 1, \cdots, N_c$, $j = 1, \cdots, N_r$, and $m = 1, \cdots, N_e$.

Then, the i-th frequency-domain control input $\boldsymbol{U}_i(k)$ as:

$$\boldsymbol{U}_i(k) = diag\{\Sigma_{j=1}^{N_r} \tilde{\boldsymbol{c}}_{ij}(k)\boldsymbol{R}_j(k)\}$$

The auxiliary signal matrix $\boldsymbol{X}(k)$ is composed by the Kronecker product of $\tilde{\boldsymbol{k}}_{mi}(k)$ and $\boldsymbol{R}_j(k)$ as follows:

$$\begin{aligned}
\boldsymbol{X}_m(k) &= [diag\{\tilde{\boldsymbol{k}}_{m1}(k)\boldsymbol{R}_1(k)\}, \cdots, diag\{\tilde{\boldsymbol{k}}_{mN_c}(k)\boldsymbol{R}_{N_r}(k)\}]^T \\
\boldsymbol{X}(k) &= [\boldsymbol{X}_1(k), \boldsymbol{X}_2(k), \cdots, \boldsymbol{X}_{N_e}(k)]
\end{aligned}$$

Next, $\boldsymbol{V}_K(k)$ and $\boldsymbol{V}_C(k)$ are defined as:

$$\boldsymbol{V}_K(k) = \tilde{\boldsymbol{K}}(k)\boldsymbol{U}(k), \quad \boldsymbol{V}_C(k) = \{[vec(\tilde{\boldsymbol{C}}(k))]\boldsymbol{X}(k)\}^T$$

Thus, the m-th auxiliary signals of the f-th frequency bin at the k-th iteration respectively are written as

$$v_{K,m}^{(f)}(k) = \Sigma_{i=1}^{N_c}\Sigma_{j=1}^{N_r} \tilde{k}_{mi}^{(f)}(k)\tilde{c}_{ij}^{(f)}(k)R_j^{(f)}(k) \tag{8}$$

$$v_{C,m}^{(f)}(k) = \Sigma_{i=1}^{N_c}\Sigma_{j=1}^{N_r} \tilde{c}_{ij}^{(f)}(k)\tilde{k}_{mi}^{(f)}(k)R_j^{(f)}(k) \tag{9}$$

where $\tilde{k}_{mi}^{(f)}(k)$, $\tilde{c}_{ij}^{(f)}(k)$, $R_j^{(f)}(k)$ are the weight of the each f-th frequency bin respectively. Since they are scalar values, we can exchange the product of two $\tilde{\boldsymbol{k}}_{mi}(k)$ and $\tilde{\boldsymbol{c}}_{ij}(k)$, and then we can establish that $\boldsymbol{v}_{K,m}(k) = \boldsymbol{v}_{C,m}(k)$ from (8) and (9).

The canceling error $e_m(n)$ detected in m-th error microphone is expressed by

$$e_m(kM+t) = d_m(kM+t) - y_m(kM+t),\ \ t = 0, 1. \cdots, M-1$$

And the m-th virtual error vectors are expressed by:

$$\boldsymbol{e}_{A,m}(k) = \boldsymbol{e}_m(k) + \{\boldsymbol{v}_{K,m}(k) - \boldsymbol{v}_{D,m}(k)\}_{last} \tag{10}$$

$$\boldsymbol{e}_{B,m}(k) = \{\boldsymbol{v}_{D,m}(k) - \boldsymbol{v}_{C,m}(k)\}_{last} \tag{11}$$

where $\{\boldsymbol{A}\}_{last}$ denotes last M elements of $IFFT\,\{\boldsymbol{A}\}$.

Then we consider the sum of two virtual errors in Fig.2 from (8)$\sim$(11) as

$$\boldsymbol{e}_{A,m}(k) + \boldsymbol{e}_{B,m}(k) = \boldsymbol{e}_m(k) + \{\boldsymbol{v}_{K,m}(k) - \boldsymbol{v}_{C,m}(k)\}_{last} = \boldsymbol{e}_m(k)$$

If the coefficient parameters in the three adaptive filters $\tilde{\boldsymbol{C}}(k)$, $\tilde{\boldsymbol{K}}(k)$, and $\tilde{\boldsymbol{D}}(k)$ so that the errors $\boldsymbol{e}_A(k)$ and $\boldsymbol{e}_B(k)$ may become zero, the canceling error vector $\boldsymbol{e}(k)$ can also converge to zero.

3.2 Adaptation for frequency-domain technique

First, we can transform the first error vector $\boldsymbol{e}_{A,m}(k)$ into $\boldsymbol{E}_{A,m}(k)$ expressed in frequency-domain using the FFT.

$$\boldsymbol{E}_{A,m}(k) = FFT\,[\ \ \boldsymbol{0} \quad \boldsymbol{e}_{A,m}(k)\ \ ]$$

where $\boldsymbol{0}$ is an M-by-1 null vector. Then, the power spectral density of the two reference signals for $\tilde{\boldsymbol{K}}(k)$ and $\tilde{\boldsymbol{D}}(k)$ in the f-th frequency bin at the k-th iteration are expressed each as

$$\begin{aligned}
\tilde{S}_{K,uu}^{(f)}(k) &= \gamma\tilde{S}_{K,uu}^{(f)}(k-1) + (1-\gamma)\{\Sigma_{i=1}^{N_c}|U_i^{(f)}(k)|^2\} \\
\tilde{S}_{D,rr}^{(f)}(k) &= \gamma\tilde{S}_{D,rr}^{(f)}(k-1) + (1-\gamma)\{\Sigma_{j=1}^{N_r}|R_j^{(f)}(k)|^2\}
\end{aligned}$$

where $f = 0, \cdots, N-1$ and γ is forgetting factor. Thus, the f-th convergence coefficients are expressed by

$$\tilde{\alpha}_K^{(f)}(k) = \alpha_K[\tilde{S}_{K,uu}^{(f)}(k)]^{-1}, \quad \tilde{\alpha}_D^{(f)}(k) = \alpha_D[\tilde{S}_{D,rr}^{(f)}(k)]^{-1}$$

where α_K and α_D are the constant convergence coefficients. Therefore, the modifications of the frequency-domain algorithm can be written as

$$\begin{aligned}
\tilde{k}_{mi}^{(f)}(k+1) &= \tilde{k}_{mi}^{(f)}(k) - \{\tilde{\alpha}_K^{(f)}(k)U_i^{(f)^H}(k)E_{A,m}^{(f)}(k)\}_{first} \\
\tilde{d}_{mj}^{(f)}(k+1) &= \tilde{d}_{mj}^{(f)}(k) + \{\tilde{\alpha}_D^{(f)}(k)R_j^{(f)^H}(k)E_{A,m}^{(f)}(k)\}_{first}
\end{aligned}$$

where $\{\boldsymbol{A}\}_{first}$ denotes first M elements of $IFFT\,\{\boldsymbol{A}\}$.

Next, we can transform the second error vector $\boldsymbol{e}_{B,m}(k)$ into $\boldsymbol{E}_{B,m}(k)$ expressed in frequency-domain using the FFT.

$$\boldsymbol{E}_{B,m}(k) = FFT\,[\ \ \boldsymbol{0} \quad \boldsymbol{e}_{B,m}(k)\ \ ]$$

Then, unlike the modifications of $\tilde{\boldsymbol{K}}(k)$ and $\tilde{\boldsymbol{C}}(k)$, the power spectral density of the reference signal for $\tilde{\boldsymbol{c}}_{ij}(k)$ in the f-th frequency bin at the k-th iteration are expressed as

$$\begin{aligned}
\tilde{P}_{C,ij}^{(f)}(k) &= \gamma\tilde{P}_{C,ij}^{(f)}(k-1) + (1-\gamma)\{\Sigma_{m=1}^{N_e}|X_{mij}^{(f)}(k)|^2\} \\
\boldsymbol{Q}_{C,ij}(k) &= diag\{[(\tilde{P}_{C,ij}^{(0)})^{-1}(k), \cdots, (\tilde{P}_{C,ij}^{(N-1)})^{-1}(k)]^T\}
\end{aligned}$$

Therefore, the modification of the frequency-domain LMS algorithm can be written as

$$\begin{aligned}
\tilde{\boldsymbol{c}}_{ij}(k+1) &= \tilde{\boldsymbol{c}}_{ij}(k) + \alpha_C FFT\,[\ \ \boldsymbol{\phi}_{C,ij}(k) \quad \boldsymbol{0}\ \ ] \\
\boldsymbol{\phi}_{C,ij}(k) &= \{[\Sigma_{m=1}^{N_e}\boldsymbol{E}_{B,m}\boldsymbol{x}_{mij}^H\boldsymbol{Q}_{C,ij}]\}_{first}
\end{aligned}$$

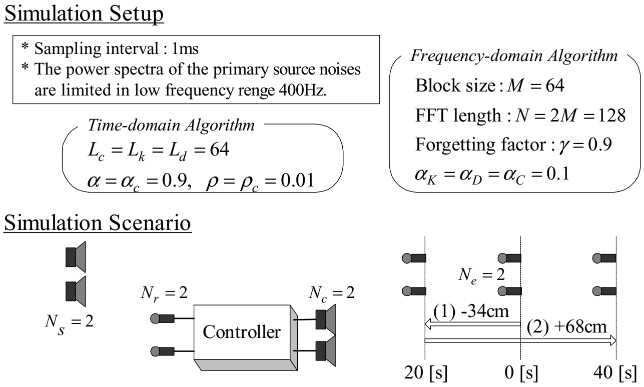


Figure 3: Simulation setup and simulation scenario

4. COMPUTATIONAL COMPLEXITY

Comparison of the computational complexity is based on the total number of multiplications involved in frequency-domain and time-domain implementations for a block size M. Consider first time-domain algorithm, with M tap weights operating on real data. Here, N_c, N_r and N_e are set to a to simplify the problem. In this case, $(2a^3+3a^2)M$ multiplications are performed to compute the output, and $(4a^3+6a^2)M+4a+6$ multiplications are performed to update the tap weights. Now consider the frequency-domain algorithm. Each N-point FFT (and IFFT) requires approximately $N\log_2 N$ real multiplications, where $N=2M$. There are $6a^2+6a$ frequency transformations performed in the proposed algorithm, which therefore account for $(6a^2+6a)N\log_2 N$ multiplications. In addition, $(2a^3+3a^2)N$ multiplications are performed to compute the output, and $(5a^3+14a^2)N$ multiplications are performed to update. The complexity ratio for the frequency-domain algorithm to time-domain algorithm is therefore,

$$\frac{10a^3+28a^2+(12a^2+12a)\log_2 M}{(6a^3+9a^2)M+4a+6}$$

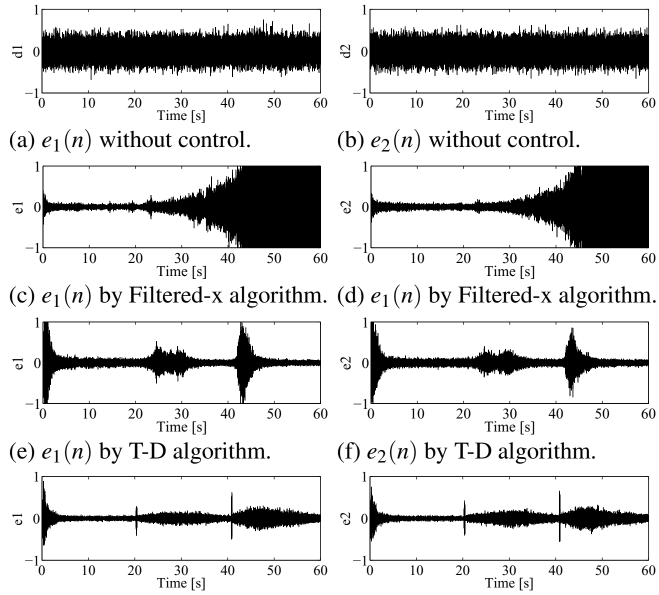
For example, for $a=2, M=1024$, use this equation shows that the frequency-domain algorithm is roughly 94 times faster than the time-domain algorithm in computational terms.

5. NUMERICAL SIMULATION

We examine the effectiveness of the multichannel frequency-domain direct adaptive algorithm in two channel ANC in a room. Sound reflections on the room walls are assumed to be suppressed in a passive manner. We used the path models that were obtained experimentally.

Fig.3 shows the simulation setup and the scenario in which the location of the two error microphones is changed by 34cm instantaneously near to the position of the primary sources by using the switches at 20 s after the start of control, and then that is changed by 68cm instantaneously far from there by using the switches at 40 s after.

Figs. 4(a) and 4(b) show the actual canceling errors $e_1(n)$ and $e_2(n)$ in a case without control. The movement of error microphones causes uncertain changes of the primary and secondary path dynamics, and so the filtered-x types of algorithm could not keep stable attenuation performance at the



(a) $e_1(n)$ without control. (b) $e_2(n)$ without control.

(c) $e_1(n)$ by Filtered-x algorithm. (d) $e_1(n)$ by Filtered-x algorithm.

(e) $e_1(n)$ by T-D algorithm. (f) $e_2(n)$ by T-D algorithm.

(g) $e_1(n)$ by F-D algorithm. (h) $e_2(n)$ by F-D algorithm.

Figure 4: Comparison of control results

first switched time as shown in Figs.4(c) and 4(d), since it needs precise knowledge on the channels dynamics. On the other hand, the direct method types of both algorithms can still attain the stable control performance even if the secondary channels change very rapidly as given in Figs.4(e) to 4(h).

6. CONCLUSION

We have presented the new frequency-domain direct adaptive algorithm for tuning the feedforward inverse controller in multichannel cases, which is effective even when the secondary path matrices are uncertain or unknown. The proposed algorithms do not need explicit identification of uncertain paths, and frequency-domain technique has the greatest computational savings. The effectiveness has been validated by numerical simulations of two-channel ANC.

REFERENCES

[1] S. J. Elliot and P. A. Nelson, *Active Noise Control*. Academic Press, 1993.

[2] S. M. Kuo and D. R. Morgan, *Active Noise Control Systems*. John Wiley and Sons, 1996.

[3] O. Tohki and S.Veres, *Active Sound and Vibration Control*, *IEE*, 2002.

[4] E. Bjarnason, “Analysis of the filtered-x lms algorithm,” *IEEE Trans. Speech and Audio Processing*, vol. 3, pp. 504–514, 1995.

[5] F. Jiang, H. Tsuji, H. Ohmori and A. Sano, “Adaptaion for active noise control,” *IEEE Control Systems*, vol. 17, pp. 36–47, 1997.

[6] M. Kamata and A. Sano, “New direct adaptive algorithm for multichannel active noise control,” in *Proc. EUSIPCO 2004*, Vienna, Austria, Sept. 6-10. 2004.

[7] J. J. Shynk, “Frequency-domain and multirate adaptive filtering,” *IEEE Signal Processing Magagine*, vol. 9, pp. 14–37, 1992.